

IMPLEMENTASI APPLICATION PROGRAMMING INTERFACE (API) UNTUK APLIKASI PRESENSI MOBILE BERBASIS FLUTTER (STUDI KASUS: RUMAH SAKIT RAJAWALI CITRA JOGJA)

Miqdad Dzaki Nashiruddin, Arridho Ramadhan Firdaus*

Teknologi Informasi, Universitas Aisyiyah Yogyakarta

Jl. Siliwangi No. 63, Mlangi, Nogotirto, Gamping, Sleman, Yogyakarta, Indonesia

arridhoramadhanfirdaus@unisayogya.ac.id

ABSTRAK

Rumah Sakit Rajawali Citra telah menerapkan sistem presensi berbasis website sebagai sarana pencatatan kehadiran pegawai. Sistem tersebut masih memiliki keterbatasan dalam mendukung mobilitas dan fleksibilitas kerja pegawai, sehingga pengembangan ke *platform mobile* menjadi kebutuhan. Penelitian ini bertujuan untuk mengimplementasikan *Application Programming Interface (API)* berbasis *framework* Laravel sebagai penghubung antara basis data sistem presensi website dengan aplikasi presensi *mobile* berbasis Flutter. Metode penelitian yang digunakan adalah *System Development Life Cycle (SDLC)* model *Waterfall* yang meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. API dirancang menggunakan arsitektur *RESTful* untuk mendukung fungsionalitas presensi masuk dan presensi pulang dengan validasi lokasi berbasis *geofencing* serta penyajian riwayat presensi pegawai. Pengujian fungsionalitas dilakukan menggunakan metode *black-box testing* dengan bantuan Postman. Hasil pengujian menunjukkan bahwa seluruh *endpoint* API berfungsi sesuai dengan kebutuhan sistem, mampu memproses data presensi secara akurat, serta melakukan verifikasi lokasi secara *real-time* dan aman. Implementasi API ini mendukung integrasi sistem presensi website dengan aplikasi *mobile* serta meningkatkan efisiensi proses pencatatan kehadiran pegawai.

Kata kunci : *Laravel, API, RESTful, Flutter, Presensi Mobile.*

1. PENDAHULUAN

Sistem pencatatan kehadiran atau presensi merupakan komponen penting dalam manajemen sumber daya manusia pada suatu institusi, termasuk rumah sakit, karena digunakan sebagai dasar evaluasi kedisiplinan dan kinerja pegawai. Rumah Sakit Rajawali Citra telah menerapkan sistem presensi berbasis *website* sebagai sarana pencatatan kehadiran pegawai. Sistem tersebut telah berjalan dengan baik dalam mendukung proses administrasi, akan tetapi masih memiliki keterbatasan dalam hal mobilitas dan fleksibilitas penggunaan, khususnya bagi pegawai dengan pola kerja dinamis dan berpindah lokasi [1].

Perkembangan teknologi *mobile* mendorong institusi untuk mengadopsi sistem presensi berbasis aplikasi *mobile* guna meningkatkan efisiensi dan kemudahan akses pencatatan kehadiran [2]. Pengembangan aplikasi *mobile* yang terpisah dari sistem *website* berpotensi menimbulkan permasalahan integrasi data, seperti inkonsistensi informasi presensi dan duplikasi penyimpanan data. Kondisi tersebut memerlukan mekanisme komunikasi data yang terstandar agar aplikasi *mobile* dapat terhubung secara langsung dengan basis data sistem presensi yang telah berjalan. Mekanisme komunikasi tersebut direalisasikan melalui *Application Programming Interface (API)* yang berfungsi sebagai penghubung antara aplikasi *mobile* dan server [3].

Arsitektur *Representational State Transfer (RESTful)* banyak digunakan dalam pengembangan API karena bersifat *stateless*, efisien, serta mudah diintegrasikan dengan berbagai platform [4]. *Framework* Laravel dipilih dalam pengembangan API

karena menyediakan arsitektur *Model-View-Controller (MVC)*, sistem *routing* yang terstruktur, serta fitur keamanan yang mendukung implementasi *RESTful API* secara optimal [5]. Selain aspek integrasi data, sistem presensi *mobile* juga perlu dilengkapi dengan validasi lokasi berbasis *geofencing* untuk memastikan bahwa pencatatan kehadiran hanya dapat dilakukan pada area kerja yang telah ditentukan [6].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengimplementasikan API *RESTful* berbasis Laravel yang terintegrasi dengan database sistem presensi website Rumah Sakit Rajawali Citra serta mendukung aplikasi presensi *mobile* berbasis Flutter. Rencana penyelesaian masalah dilakukan melalui perancangan dan implementasi API yang mendukung fungsionalitas presensi masuk dan presensi pulang dengan validasi lokasi menggunakan Formula Haversine di sisi server, serta pengujian fungsionalitas menggunakan metode *black-box testing*. Implementasi ini diharapkan mampu meningkatkan efisiensi, keamanan, dan keandalan sistem presensi pegawai berbasis *mobile*.

2. TINJAUAN PUSTAKA

2.1. Application Programming Interface (API)

Application Programming Interface (API) merupakan sekumpulan aturan dan protokol yang memungkinkan suatu aplikasi berkomunikasi dengan aplikasi lain atau dengan sistem backend. API berfungsi sebagai perantara antara klien dan server dalam sistem terdistribusi [3]. *RESTful API* memanfaatkan protokol HTTP dengan metode standar seperti GET dan POST serta menggunakan format data

JSON yang ringan dan mudah diproses oleh berbagai platform [4].

Penggunaan RESTful API banyak diterapkan pada pengembangan aplikasi modern karena kemampuannya dalam mendukung komunikasi lintas platform. Prinsip stateless pada REST membuat setiap permintaan dari klien bersifat independen, sehingga meningkatkan skalabilitas dan performa sistem. Selain itu, penggunaan HTTP status code seperti 200 (OK), 201 (Created), dan 401 (Unauthorized) membantu klien dalam memahami hasil pemrosesan permintaan secara lebih jelas dan terstandarisasi.

2.2. Framework Laravel

Laravel adalah framework PHP berbasis *Model-View-Controller* (MVC) yang banyak digunakan dalam pengembangan aplikasi web dan RESTful API. Laravel menyediakan fitur routing, middleware, ORM Eloquent, serta mekanisme keamanan yang memudahkan pengelolaan autentikasi dan otorisasi pengguna [11]. Laravel Sanctum mendukung autentikasi berbasis token yang sesuai untuk aplikasi mobile dan API.

Laravel juga mendukung pengelolaan API melalui mekanisme middleware yang memungkinkan pengembang membatasi akses ke *endpoint* tertentu. Dengan adanya middleware autentikasi, hanya pengguna yang memiliki token valid yang dapat mengakses layanan API. Hal ini sangat penting dalam sistem presensi karena data kehadiran bersifat sensitif dan harus terlindungi dari akses yang tidak sah.

2.3. Aplikasi Mobile Berbasis Flutter

Flutter merupakan framework *cross-platform* yang dikembangkan oleh Google untuk membangun aplikasi *mobile* dengan satu basis kode. Flutter menggunakan bahasa pemrograman *dart* dan mampu menghasilkan aplikasi dengan performa yang mendekati *native* [1]. Integrasi Flutter dengan RESTful API memungkinkan aplikasi *mobile* mengakses data presensi secara *real-time*.

Keunggulan Flutter dalam pengembangan aplikasi presensi terletak pada kemampuannya dalam mengakses fitur perangkat seperti GPS, kamera, dan konektivitas jaringan secara optimal. Dengan satu basis kode, aplikasi dapat dijalankan pada sistem operasi Android maupun iOS, sehingga mengurangi biaya dan waktu pengembangan. Flutter juga mendukung komunikasi asinkron dengan RESTful API, yang memungkinkan pertukaran data presensi dilakukan secara cepat dan *real-time*.

2.4. Sistem Presensi Berbasis Lokasi (Geofencing)

Geofencing adalah metode pembatasan area virtual berdasarkan koordinat geografis tertentu. Presensi hanya dapat dilakukan jika pengguna berada dalam radius yang ditentukan [6]. Formula Haversine digunakan untuk menghitung jarak dua titik berdasarkan lintang dan bujur secara akurat [7].

Perhitungan jarak di sisi server dinilai lebih aman dibandingkan di sisi klien [10].

Penerapan validasi lokasi pada sistem presensi bertujuan untuk meminimalkan kecurangan, seperti melakukan presensi di luar area kerja. Validasi lokasi yang dilakukan di sisi server dinilai lebih aman karena mengurangi risiko manipulasi koordinat GPS oleh pengguna. Dengan demikian, sistem presensi menjadi lebih andal dan akurat dalam mencatat kehadiran pegawai.

2.5. Penelitian Terkait

Beberapa penelitian sebelumnya telah membahas sistem presensi berbasis *mobile* dan API. Penelitian oleh Saputra dan Asmiatun [1] mengimplementasikan presensi berbasis Flutter dengan pemanfaatan lokasi pengguna. Penelitian Ardhana *et al.* [3] membahas pengembangan RESTful API menggunakan Laravel, namun tidak difokuskan pada integrasi sistem presensi berbasis lokasi. Perbedaan penelitian ini terletak pada integrasi API dengan sistem presensi website yang telah berjalan serta validasi lokasi di sisi server.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan Penelitian ini menggunakan pendekatan kualitatif-terapan dengan metodologi System Development Life Cycle (SDLC) model Waterfall. Model ini dipilih karena memberikan alur kerja yang sistematis, di mana setiap tahapan harus diselesaikan secara berurutan sebelum melangkah ke tahap berikutnya, sangat sesuai untuk pengembangan API yang membutuhkan akurasi data yang tinggi.

3.1. Analisis Kebutuhan (Requirement Analysis)

Tahap ini merupakan fondasi untuk memahami bagaimana sistem website yang sudah ada bekerja. Peneliti melakukan identifikasi melalui:

- Observasi Basis Data: Menganalisis struktur tabel pada database MySQL milik Rumah Sakit Rajawali Citra untuk memastikan API dapat mengakses data tanpa merusak integritas sistem website yang berjalan.
- Identifikasi Fitur: Menentukan fungsi-fungsi krusial yang dibutuhkan oleh pegawai, seperti sistem masuk (*check-in*), sistem pulang (*check-out*), dan pemantauan riwayat kehadiran secara mandiri melalui ponsel.
- Spesifikasi Teknis: Menetapkan standar komunikasi data menggunakan format JSON agar aplikasi Flutter dapat memproses informasi dengan ringan dan cepat.

3.2. Perancangan Sistem (System Design)

Peneliti merancang "cetak biru" sistem sebelum masuk ke tahap pengkodean.

- Arsitektur RESTful: API dirancang menggunakan arsitektur REST agar setiap permintaan (*request*) bersifat mandiri (*stateless*)

dan efisien dalam penggunaan sumber daya server.

- b. Logika Geofencing: Merancang skenario pembatasan area virtual. Peneliti menetapkan titik koordinat lokasi rumah sakit sebagai titik acuan tetap.
- c. Keamanan Data: Merancang sistem autentikasi menggunakan Laravel Sanctum, di mana setiap pengguna akan mendapatkan "token digital" unik setelah login sebagai syarat untuk mengakses fitur presensi.

3.3. Implementasi (Implementation)

Pada tahap ini, rancangan teknis diwujudkan dalam bentuk baris kode program.

- a. Backend Development: Menggunakan bahasa pemrograman PHP 8.2 dan *framework* Laravel 12 untuk membangun logika bisnis di sisi server.
- b. Integrasi Formula Haversine: Mengkodekan rumus matematika Haversine ke dalam sistem untuk menghitung jarak antara koordinat GPS ponsel pegawai dengan koordinat rumah sakit secara otomatis.
- c. Database Connection: Menghubungkan API dengan server database MySQL sehingga data yang dimasukkan melalui aplikasi mobile akan langsung tersinkronisasi dengan database pusat rumah sakit.

3.4. Pengujian (Testing)

Tahap akhir adalah memastikan sistem berjalan tanpa kesalahan menggunakan metode *Black-Box Testing*.

- a. Simulasi Penggunaan: Peneliti menggunakan aplikasi Postman untuk mensimulasikan berbagai kondisi, seperti mencoba absen saat berada jauh

dari rumah sakit atau mencoba login dengan akun yang tidak terdaftar.

- b. Validasi Output: Memastikan server selalu memberikan respon yang tepat, seperti kode "200 OK" untuk permintaan yang berhasil atau "201 Created" saat data presensi baru berhasil disimpan.

Pemilihan model Waterfall dalam penelitian ini didasarkan pada kebutuhan sistem yang memiliki spesifikasi jelas sejak awal dan tidak mengalami perubahan signifikan selama proses pengembangan. Setiap tahapan dalam model Waterfall menghasilkan keluaran yang menjadi dasar bagi tahap berikutnya, sehingga meminimalkan kesalahan dalam pengembangan API presensi yang terintegrasi dengan sistem website eksisting.

Selain itu, pendekatan ini memudahkan proses dokumentasi sistem, mulai dari analisis kebutuhan hingga tahap pengujian. Dokumentasi yang baik sangat diperlukan untuk mendukung proses pemeliharaan dan pengembangan lanjutan sistem presensi di masa mendatang.

4. HASIL DAN PEMBAHASAN

4.1. Perancangan Endpoint RESTful API

Perancangan API mengikuti prinsip *Representational State Transfer* (REST), di mana setiap sumber daya diakses melalui URL unik dengan metode HTTP standar seperti GET dan POST. Autentikasi pengguna diterapkan menggunakan Laravel Sanctum untuk memastikan keamanan akses API [9]. Endpoint yang dibangun difokuskan pada kebutuhan aplikasi presensi mobile yang ditunjukkan pada Tabel 1.

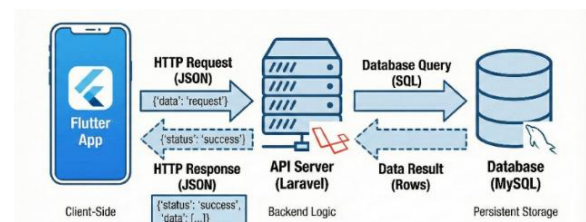
Tabel 1. Daftar Endpoint Utama API Presensi

Endpoint URL	Metode HTTP	Fungsi	Keterangan
/api/login	POST	Otentikasi Pengguna	Mengembalikan token Sanctum jika login berhasil
/api/user	GET	Data Pegawai	Mengambil data pegawai yang sedang login
/api/checkin	POST	Presensi Masuk	Mencatat presensi dengan validasi lokasi (<i>geofencing</i>)
/api/checkout	POST	Presensi Pulang	Mencatat presensi pulang dengan validasi lokasi
/api/history	GET	Riwayat Presensi	Menampilkan riwayat presensi pegawai

Berdasarkan Tabel 1, endpoint API dirancang untuk memenuhi seluruh kebutuhan fungsional aplikasi presensi mobile. Penggunaan format data JSON pada setiap respons API mempermudah aplikasi Flutter dalam memproses dan menampilkan data presensi secara efisien.

4.2. Arsitektur Sistem

Arsitektur sistem presensi mobile terdiri dari tiga komponen utama, yaitu aplikasi mobile Flutter sebagai klien, RESTful API berbasis Laravel sebagai server, dan basis data MySQL sebagai penyimpanan data presensi. Aplikasi Flutter berkomunikasi dengan API melalui protokol HTTP dengan format data JSON.



Gambar 1. Arsitektur Komunikasi Aplikasi Presensi Mobile

Gambar 1, API berperan sebagai penghubung utama antara aplikasi mobile dan basis data. Seluruh logika bisnis, autentikasi, dan validasi lokasi diproses di sisi server, sehingga meningkatkan keamanan dan konsistensi data presensi.

4.3. Implementasi Validasi Geofencing

Validasi lokasi (*geofencing*) diterapkan untuk memastikan bahwa pegawai hanya dapat melakukan presensi ketika berada dalam radius 70 meter dari titik koordinat Rumah Sakit Rajawali Citra. Perhitungan jarak dilakukan menggunakan Formula Haversine yang menghitung jarak dua titik berdasarkan nilai lintang dan bujur [7]. Formula Haversine ditunjukkan pada rumus berikut.

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

Keterangan:

- d adalah jarak
- r adalah jari-jari bumi

- ϕ adalah garis lintang (latitude)
- λ adalah garis bujur (longitude).

Hasil perhitungan jarak dibandingkan dengan nilai radius yang telah ditentukan. Jika jarak pengguna berada dalam radius 70 meter, maka presensi dicatat sebagai presensi dalam zona, sedangkan jika berada di luar radius, presensi akan dicatat luar zona.

4.4. Hasil Pengujian Fungsionalitas API

Pengujian fungsionalitas dilakukan menggunakan metode *Black-Box Testing* dengan bantuan aplikasi Postman untuk memastikan seluruh endpoint API berjalan sesuai dengan kebutuhan sistem [8]. Endpoint API ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pengujian Endpoint API Presensi

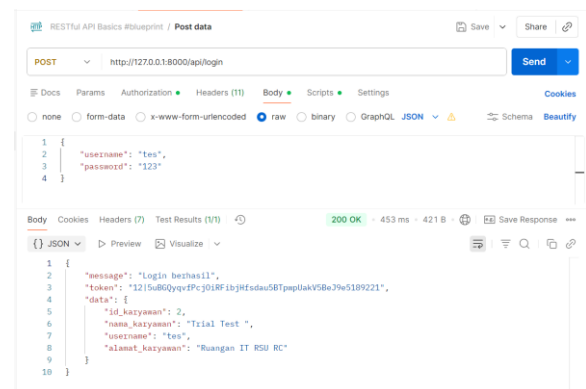
Endpoint	Data Input	Hasil yang Diharapkan	Hasil Aktual	Status
/api/login	Username dan password valid	Status Code 200, token Sanctum dikembalikan	Status Code 200, token diterima	Berhasil
/api/checkin	ID shift, koordinat dalam radius	Status Code 201, presensi masuk berhasil	Status Code 201, presensi tercatat	Berhasil
/api/checkout	Koordinat dalam radius	Status Code 200, presensi pulang berhasil	Status Code 200, presensi pulang berhasil	Berhasil
/api/user	Token Sanctum	Status Code 200, data pegawai ditampilkan	Status Code 200, data JSON diterima	Berhasil
/api/history	Token Sanctum	Status Code 200, riwayat presensi ditampilkan	Status Code 200, data JSON diterima	Berhasil

Hasil pengujian menunjukkan bahwa seluruh endpoint API mampu menangani permintaan dengan baik dan konsisten. Keberhasilan autentikasi menggunakan Laravel Sanctum memastikan bahwa hanya pengguna yang memiliki token valid yang dapat mengakses layanan presensi. Hal ini meningkatkan keamanan sistem serta mencegah akses tidak sah terhadap data presensi pegawai.

Selain aspek fungsionalitas, performa API juga menunjukkan hasil yang baik dengan waktu respons yang relatif cepat. Waktu respons di bawah 300 ms menunjukkan bahwa API mampu mendukung kebutuhan aplikasi presensi mobile yang bersifat *real-time*. Dengan performa tersebut, aplikasi dapat digunakan secara nyaman tanpa menimbulkan keterlambatan dalam proses presensi.

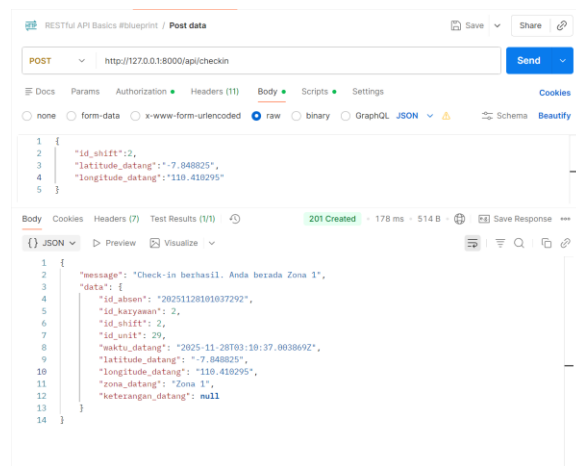
4.5. Dokumentasi Pengujian Menggunakan Postman

Pengujian API dilakukan dengan mensimulasikan permintaan dari aplikasi *mobile* menggunakan Postman.



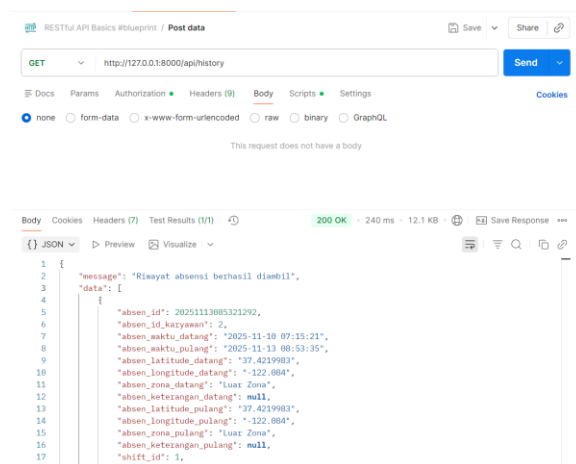
Gambar 2. Pengujian Endpoint Login

Gambar 2 menampilkan hasil pengujian *endpoint login* menggunakan Postman untuk memverifikasi proses autentikasi pengguna. Pengujian dilakukan dengan mengirimkan kredensial berupa *username* dan *password* ke server melalui metode POST. Hasil pengujian menunjukkan bahwa sistem berhasil memvalidasi data pengguna dan mengembalikan token autentikasi Laravel Sanctum dengan status kode 200, yang menandakan proses login berjalan dengan sukses dan aman.



Gambar 3. Pengujian Endpoint Presensi Masuk

Gambar 3 menunjukkan hasil pengujian *endpoint* presensi masuk yang dilakukan menggunakan Postman. Permintaan dikirim dengan menyertakan data koordinat lokasi pengguna untuk memastikan bahwa sistem dapat melakukan validasi *geofencing* dengan benar. Hasil pengujian membuktikan bahwa API mampu memverifikasi lokasi pengguna menggunakan Formula Haversine dan mencatat presensi masuk dengan status kode 201 ketika pengguna berada dalam radius yang ditentukan.



Gambar 4. Pengujian Endpoint Riwayat Presensi

Gambar 4 menampilkan hasil pengujian *endpoint* riwayat presensi yang bertujuan untuk menampilkan data kehadiran pegawai. Permintaan dikirim dengan menyertakan token autentikasi yang valid untuk memastikan keamanan akses data. Hasil pengujian menunjukkan bahwa API berhasil mengembalikan data riwayat presensi dalam format JSON dengan status kode 200, yang menandakan bahwa proses pengambilan data berjalan dengan baik dan sesuai dengan kebutuhan sistem.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan API berbasis Laravel dengan arsitektur RESTful untuk mendukung aplikasi presensi mobile berbasis Flutter

di Rumah Sakit Rajawali Citra. API mampu menangani autentikasi pengguna, pencatatan presensi masuk dan pulang dengan validasi lokasi menggunakan Formula Haversine, serta penyajian riwayat presensi. Hasil pengujian menunjukkan seluruh *endpoint* berfungsi dengan baik dan memiliki waktu respons yang optimal. Pengembangan selanjutnya disarankan menambahkan fitur notifikasi *real-time* dan monitoring penggunaan API.

DAFTAR PUSTAKA

- [1] A. Saputra and S. Asmiatun, "Penerapan Geotagging pada Sistem Presensi Berbasis Lokasi Menggunakan Flutter," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 4, pp. 789–798, 2023.
- [2] K. Komarudin, P. Purwantoro, and A. Suharso, "Aplikasi Presensi Menggunakan Geolocation Berbasis Mobile Framework Flutter (Studi Kasus: Madrasah Aliyyah Negeri 2 Karawang)," *Jurnal Informasi dan Komputer*, vol. 11, no. 2, 2023.
- [3] V. Y. P. Ardhana, M. T. Hidayat, M. Jannah, S. Sumiati, P. Rini, and N. Sari, "Implementasi RESTful API pada Laravel dan Simulator IoT Wokwi untuk Pengukuran Suhu dan Kelembaban Menggunakan Metode Waterfall," *Arcitech: Journal of Computer Science and Artificial Intelligence*, vol. 3, no. 2, pp. 93–109, 2023.
- [4] M. A. Biruni, A. Faisol, and N. Vendyansyah, "Penerapan REST API dan Integrasi Midtrans Sebagai Payment Gateway Pada Platform Pelatihan Online," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1, pp. 12–18, 2023.
- [5] K. S. Utami, N. P. Sastra, and D. M. Wiharta, "Pengembangan Metode Autentikasi pada Sistem Presensi Berbasis Aplikasi Mobile," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, pp. 615–623, 2021.
- [6] H. Harifin, "Aplikasi Presensi Siswa Berbasis Location Based Services (LBS) Dengan Haversine Formula di SMK Islam Al-Futuhiyyah," *Indonesian Journal of Applied Informatics (IJAI)*, vol. 8, no. 2, 2024.
- [7] B. P. Ilham, "Pembatasan Jarak Presensi dalam Sistem Presensi Daring Menggunakan Formula Haversine," *TeIKa: Jurnal Teknologi Informasi dan Komunikasi*, vol. 14, no. 2, pp. 117–127, 2024.
- [8] R. G. Guntara and V. Azkarin, "Implementasi dan Pengujian REST API Sistem Reservasi Ruang Rapat Dengan Metode Black Box Testing," *Jurnal Minfo Polgan*, vol. 12, no. 1, pp. 1229–1238, 2023.
- [9] N. Fauzi and M. Hardjianto, "Implementasi Web Service RESTful API pada Aplikasi ShamoStore Berbasis Android Menggunakan Flutter dan Laravel Sanctum," in *Seminar Nasional*

- Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, 2022, pp. 1147–1154.
- [10] M. I. Bahtiar, R. Helilintar, and D. W. Widodo, “Sistem Presensi Menggunakan Algoritma Haversine Dengan Global Positioning System,” in *Prosiding Seminar Nasional Inovasi Teknologi (SEMNAS INOTEK)*, vol. 8, pp. 1035–1043, 2024.
- [11] H. F. Herdiyatomoko, “Desain Sistem Backend Berbasis REST API Menggunakan Framework Laravel 7,” *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 2, pp. 136–144, 2022.
- [12] H. Mutaqin and N. Nurhayati, “Perancangan Aplikasi Absensi Menggunakan Metode Haversine Berbasis Android pada PT. Rangkai Utama Berjaya,” *Informatika: Jurnal Teknik Informatika*, vol. 11, no. 2, pp. 112–119, 2023.